Journal of Fuzzy Extension and Applications

www.journal-fea.com

J. Fuzzy. Ext. Appl. Vol. 5, No. 4 (2024) 573–593.

Paper Type: Research Paper

On the Simplex-based Methods for Neutrosophic Linear Programming Problems

Sulima Ahmed Mohammed Zubair[1,*], Najla ELzein AbuKaswi Osman[2], Wiem Abedelmonem Salah Ben Khalifa[2], Amal Hassan Mohammed Yassin[2]

[1] Department of Mathematics, College of Sciences, Qassim University, Buraydah 51452, Saudi Arabia; s.zebir@qu.edu.sa.
[2] College of Arts, Imam Abdulrahman Bin Faisal University, Saudi Arabia; neosman@iau.edu.sa; wabenkhalifa@iau.edu.sa; ahyaseen@iau.edu.sa.

Citation:

Received: 03 August 2024 *Revised: 08 October 2024* *Accepted: 17 November 2024*	Ahmed Mohammed Zubair, S., ELzein AbuKaswi Osman, N., Abedelmonem Salah Ben Khalifa, W., & Hassan Mohammed Yassin, A. (2024). On the Simplex-based methods for Neutrosophic linear programming problems. *Journal of fuzzy extension and applications, 5*(4), 573-593.

Abstract

This paper investigates Neutrosophic Linear Programming (NLP) and focuses on one of the most suitable approaches to solve it, which is called the Simplex-based model. This type of method, inspired by the classic Simplex algorithm, is in search of an optimal basic neutrosophic feasible solution, and several attractive models of it have been proposed in recent years. However, due to neutrosophic logic considers three dimensions of a problem, using a direct generalization of the simplex algorithm (which has been done in existing methods), the computational volume is greatly increased even for the small problems, and as a result, the use of these models in real-world issues will be questioned. To solve this gap, we consider NLP and propose an effective, simple model that can significantly reduce computational tasks and address these deficits in the mentioned models. Some numerical experiments with the comparison results are provided to explain the efficiency and superiority of the proposed approach.

Keywords: Linear programming, Neutrosophic linear programming, Simplex method, Triangular neutrosophic numbers, Single valued triangular neutrosophic numbers.

1 | Introduction

Today, the Linear Programming (LP) problem is not hidden from anyone due to its high efficiency in systematically optimizing real-world issues. There is also a need for uncertainty and indeterminacy environments to handle LP in these situations. This medium was done in different environments of fuzzy

Corresponding Author: s.zebir@qu.edu.sa
https://doi.org/10.22105/jfea.2024.476734.1615

sets and their extensions. For example, Zimmermann [1] has divided Fuzzy LP (FLP) issues into two categories: 1) symmetrical and 2) non-symmetrical. In a symmetrical problem, the objectives and constraints have the same weight, but in a non-symmetrical situation, the objectives and limitations are not equally important and have distinct weights. Tanaka and Asai [2] used fuzzy numbers to represent the coefficients of the deciding factors in their possibilistic LP formulation. Verdegay [3] demonstrated that the dual of an FLP issue with fuzzy cost coefficients in the objective function is an FLP problem with fuzzy inequality constraints. Najafi and Edalatpanah [4] provide an analytical approach for LP. Delgado et al. [5] have developed a multiple-goal LP strategy for solving an FLP, assuming that the parameters are Trapezoidal Fuzzy Numbers (TrFNs). for more models about FLP [6]–[16].

Wan et al. [17] introduced a new Intuitionistic Fuzzy Linear Programming (IFLP) technique for selecting logistics outsourcing service providers. Pérez-Cañedo and Concepción-Morales [18] addressed the deficiencies of the existing IFLP and offered a lexicographic criterion for ranking LR-type intuitively fuzzy numbers. In addition, they developed a strategy for unique optimal solutions to fully IFLP issues. Aliev et al. [19] suggested a new approach using differential evolution optimization to solve LP problems under Z-numbers. Ahmad and Adhami [20] established a model of LP in the context of a spherical fuzzy set. Furthermore, LP has been considered in other fuzzy set extensions such as Grey systems [21], [22], Rough sets [23], [24], Vague sets [25], [26], Hesitant fuzzy sets [27], [28], Pythagorean fuzzy sets [29]–[32], etc.

The Neutrosophic Set (NS) is one of the best environments to handle uncertainty and indeterminacy issues. The critical distinction between NS and other types of fuzzy sets is introducing the degree of indeterminacy/neutrality. This logic considers a problem from three-point of view: 1) the degrees of membership-Truth (T), 2) Indeterminacy (I), and 3) non-membership-Falsehood (F) that are independent, and their sum (as single-valued numbers) can be up to 3 [33].

Due to the realism that exists in the study of problems with this logic, NS attracted the attention of many scientists. Consequently, different versions of it, such as simplified NS [34]–[36], interval NS [37], [38], bipolar NS [39], [40], and neutrosophic structured element [41], were invented and used in various applications [42]–[50].

NSs also have been used many times in LP, and attractive models have been proposed to solve Neutrosophic Linear Programming (NLP). Preliminary works can be seen in [51], [52]. Ye [53] presented some basic concepts of Neutrosophic Numbers (NNs) and suggested some ideas to solve LP. Abdel-Basset et al. [54] considered trapezoidal NNs for solving fully NLP, which has been challenged in [55], [56]. Edalatpanah [57], using a non-linear approach, solves NLP with trapezoidal single-valued NSs; a direct method for triangular NLP can be seen in [58]. Khatter [59] introduced and extended a kind of α-cut technique and presented a model for NLP with the possibility concept. Das and Dash [60] considered the flaws of [52] and suggested a modified version of it. Khalifa and Kumar [61] presented a model and applied it to a stock portfolio problem using a ranking function. Das et al. [62] considered the duality theorem and proposed a new model that can overcome Abdel-Basset et al.'s problems [54]; a parametric attitude can be found in [63]. Ahmed [64], employing unrestricted LR-type, introduced a scheme for single-valued NLP.

All of the mentioned works are according to ranking or arithmetic approaches. Nevertheless, a third view has received much attention recently and is known as Simplex-Based Methods (SBMs). Bera and Mahapatra [65], [66] presented an SBM method for generalized trapezoidal NS in the objective function LP is NN. They also extended the dual simplex technique for NLP that, in addition to the objective function, the variables and right-hand-side vectors are also NNs [66]. Badr et al. [67] considered the situation that the initial feasible solution is not at hand and, using artificial variables in the classic two-phase simplex method, introduced a model for NLP; see also Big-M simplex technique with NN in [68], [69]. Nafei et al. [70] used SBM for an interval-valued version of NLP. El Seedy et al. [71] suggested an artificial variable-free simplex algorithm for NLP where the artificial variable is virtual and did not consider extra columns in the simplex tables; see also [72]. Kar et al. [73] also provided a type of SBM for NLP; see also [74] and [75] for duality approaches.

All of these models are interesting. However, since neutrosophic logic considers three dimensions of a problem, using a direct generalization of the simplex algorithm (done in the existing methods), the computational volume is significantly increased. Therefore, the use of these models in real-world issues will be questioned.

We consider SBM for NLP with Triangular NNs (TNNs) and Single-Valued Triangular NNs (SVTNNs) to solve this gap. We propose a practical, straightforward model that can significantly reduce computational tasks and address these deficits in the mentioned models. We demonstrate that the optimal solution of the NLP can be obtained humbly by considering the same Crisp LP (CLP) without solving any neutrosophic problems or auxiliary problems, as mentioned in [60]–[68]. Furthermore, without growing the number of constraints and parameters of the original problem, the new strategy provides a neutrosophic optimal solution corresponding to the existing ranking approaches [65]-[75]. However, it is simpler and computationally more efficient than these models. Therefore, the main contributions of the new strategy are:

I. Unlike existing neutrosophic simplex-based models (primal and dual), which always determine the entering and leaving variables by solving neutrosophic linear systems in all iterations, the new strategy finds these variables without solving any neutrosophic systems. This approach will lead to less computational complexity;

II. The existing Simplex and non-simplex methods, i.e., all the mentioned methods [65]-[75], involve many neutrosophic arithmetic operations. However, our approach offers the optimal neutrosophic solution without neutrosophic arithmetic operations.

III. Non-simplex methods (ranking approach) [65]-[75] need to add a large number of constraints as well as additional variables to the main problem, which will directly increase the computation time and complexity of the problem. However, the proposed method provides the optimal solution without additional variables or constraints.

2 | Basic Concepts

Here, we study some basic concepts needed for other sections; see [53], [69], [71] for more details.

Definition 1. $\Upsilon^{\aleph} = <(\rho^l, \rho^m, \rho^u), (\eta, \iota, f)>$ is called TNN so that T, I, and F are as follows:

$$T_{\Upsilon^{\aleph}}(x) = \begin{cases} \dfrac{\left(x-\rho^l\right)}{\left(\rho^m-\rho^l\right)}\eta, & \rho^l \le x < \rho^m, \\ \eta, & x = \rho^m, \\ \dfrac{\left(\rho^u-x\right)}{\left(\rho^u-\rho^m\right)}\eta, & \rho^m \le x < \rho^u, \\ 0, & \text{otherwise.} \end{cases}$$

$$I_{\Upsilon^{\aleph}}(x) = \begin{cases} \dfrac{\left(\rho^m-x\right)}{\left(\rho^m-\rho^l\right)}\iota, & \rho^l \le x < \rho^m, \\ \iota, & x = \rho^m, \\ \dfrac{\left(x-\rho^u\right)}{\left(\rho^u-\rho^m\right)}\iota, & \rho^m \le x < \rho^u, \\ 1, & \text{otherwise.} \end{cases}$$

$$F_{\Upsilon^{\aleph}}(x)=\begin{cases}\dfrac{\left(\rho^{m}-x\right)}{\left(\rho^{m}-\rho^{l}\right)}f, & \rho^{l}\le x<\rho^{m},\\ f, & x=\rho^{m},\\ \dfrac{\left(x-\rho^{u}\right)}{\left(\rho^{u}-\rho^{m}\right)}f, & \rho^{m}\le x<\rho^{u},\\ 1, & \text{otherwise},\end{cases}$$

where $0\le T_{\Upsilon^{\aleph}}(x)+I_{\Upsilon^{\aleph}}(x)+F_{\Upsilon^{\aleph}}(x)\le 3, x\in\Upsilon^{\aleph}$. Moreover, when $\rho^{l}\ge 0$, $\Upsilon^{\aleph}$ is called a nonnegative TNN. In addition, we define zeros as $0^{\aleph}=<(0,0,0),(1,1,1)>$.

Definition 2. For two TNNs $\Upsilon_i^{\aleph}=<(\rho_i,\nu_i,\kappa_i),(\eta_i,\iota_i,f_i)>(i=1,2)$:

I. $\Upsilon_1^{\aleph}\oplus\Upsilon_2^{\aleph}=<(\rho_1+\rho_2,\nu_1+\nu_2,\kappa_1+\kappa_2),(\eta_1\wedge\eta_2,\iota_1\vee\iota_2,f_1\vee f_2)>$.

II. $\Upsilon_1^{\aleph}-\Upsilon_2^{\aleph}=<(\rho_1-\kappa_2,\nu_1-\nu_2,\kappa_1-\rho_2),(\mu_1\wedge\mu_2,\nu_1\vee\nu_2,\omega_1\vee\omega_2)>$.

III. $\Upsilon_1^{\aleph}\otimes\Upsilon_2^{\aleph}=<(\rho_1\rho_2,\nu_1\nu_2,\kappa_1\kappa_2),(\eta_1\wedge\eta_2,\iota_1\vee\iota_2,f_1\vee f_2)>$, if $\rho_1>0,\nu_1>0$,

IV. $\alpha\Upsilon_1^{\aleph}=\begin{cases}<(\alpha\rho_1,\alpha\nu_1,\alpha\kappa_1),(\eta_1,\iota_1,f_1)>, & \text{if } \alpha>0\\ <(\alpha\kappa_1,\alpha\nu_1,\alpha\rho_1),(\eta_1,\iota_1,f_1)>, & \text{if } \alpha<0\end{cases}$.

To compare two TNNs, we can use the ranking function $\psi(.)$. It is proved that $\Upsilon_1^{\aleph}\le\Upsilon_2^{\aleph}$ iff $\psi(\Upsilon_1^{\aleph})\le\psi(\Upsilon_2^{\aleph})$. Here, we define two valid classes of ranking functions.

Definition 3. For TNN $\Upsilon^{\aleph}=<(\rho^{l},\rho^{m},\rho^{u}),(\eta,\iota,f)>$, we have

$$\psi(\Upsilon^{\aleph})=\frac{\rho^{l}+\rho^{m}+\rho^{u}}{9}(2+\eta-\iota-f).$$

Definition 4. $P^{\aleph}=<(\rho_1,l_1,\upsilon_1),(\rho_2,l_2,\upsilon_2),(\rho_3,l_3,\upsilon_3)>$ is called SVTNN so that T, I, and F are as follows:

$$T_{P^{\aleph}}(x)=\begin{cases}\dfrac{\left(x-\rho_1+l_1\right)}{l_1}, & \rho_1-l_1\le x<\rho_1,\\ 1, & x=\rho_1,\\ \dfrac{\left(\rho_1-x+\upsilon_1\right)}{\upsilon_1}, & \rho_1\le x<\rho_1+\upsilon_1,\\ 0, & \text{otherwise.}\end{cases}$$

$$I_{P^{\aleph}}(x)=\begin{cases}\dfrac{\left(\rho_2-x\right)}{l_2}, & \rho_2-l_2\le x<\rho_2,\\ 0, & x=\rho_2,\\ \dfrac{\left(x-\rho_2\right)}{\upsilon_2}, & \rho_2\le x<\rho_2+\upsilon_2,\\ 1, & \text{otherwise.}\end{cases}$$

$$F_{P^{\aleph}}(x)=\begin{cases}\dfrac{(\rho_3-x)}{l_3}, & \rho_3-l_3\le x<\rho_3,\\ 0, & x=\rho_3,\\ \dfrac{(x-\rho_3)}{\upsilon_3}, & \rho_3\le x<\rho_3+\upsilon_3,\\ 1, & \text{otherwise.}\end{cases}$$

Definition 5. For SVTNN $P^{\aleph}=<(\rho_1,l_1,\upsilon_1),(\rho_2,l_2,\upsilon_2),(\rho_3,l_3,\upsilon_3)>$, we have

$$\psi(P^{\aleph})=\frac{1}{18}[(6\rho_1-l_1+\upsilon_1)\alpha^n+2\{(6\rho_2-l_2+\upsilon_2)+(6\rho_3-l_3+\upsilon_3)\}(1-\alpha^n)],$$

where $\alpha\in[0,1], n\in R$.

3 | Problem Formulation for NLP

We begin with some basic notations and preliminary results, which we refer to later. For details, we refer to [65]–[69].

NLP with m constraints and n variables that, except for coefficients matrix, other parameters are expressed as NNs may be formulated as follows:

$$\begin{aligned}&\text{Max} \quad z^N\approx c^N x^N,\\ &\text{s.t}: \quad Ax^N\le b^N,\\ &\qquad\quad x^N\ge 0^N.\end{aligned} \tag{1}$$

With some slack variables, *Eq. (1)* can be transformed to

$$\begin{aligned}&\text{Max} \quad z^N\approx c^N x^N,\\ &\text{s.t}: \quad Ax^N\approx b^N,\\ &\qquad\quad x^N\ge 0^N,\end{aligned} \tag{2}$$

where $\text{rank}(A)=m$ and A is a $m\times n$ matrix. In addition, x^N that fulfills all the constraints of *Eq. (1)* is called the neutrosophic FS (feasible solution).

After carrying out permutations on the column of A, let $A=(B,N)$, where B is an $m\times m$ invertible matrix and is called *basic* matrix and N is an $m\times n-m$ matrix which is called Nonbasic Matrix (NBM). Similarly, the solution x^N to the equation $Ax^N=b^N$ can be arranged concerning B, N, i.e. $x^N=\begin{pmatrix}x_B^N\\ x_N^N\end{pmatrix}$, where

$$\begin{aligned}&Ax^N\approx b^N\Rightarrow(B,N)\begin{pmatrix}x_B^N\\ x_N^N\end{pmatrix}\approx b^N,\\ &\qquad\Rightarrow Bx_B^N+Nx_N^N\approx b^N,\\ &\qquad\Rightarrow x_B^N\approx B^{-1}b^N-B^{-1}Nx_N^N.\end{aligned} \tag{2}$$

Then,

$$\begin{cases}x_B^N\approx B^{-1}b^N\ge 0^N,\\ x_N^N\approx 0^N,\end{cases} \tag{3}$$

is called a Basic Solution (BS) of NLP. This BS is said to be the neutrosophic Basic Feasible Solution (BFS) of the neutrosophic system of linear equations $Ax^N\approx b^N$, when the condition $x_B^N\ge 0^N$ met. The variables

associated with BS are neutrosophic basic variables, while others x_N^N are said to be nonbasic variables. We express the basic variable from *Eqs. (1)* and *(2)* as

$$
\begin{aligned}
& x_B^N \approx B^{-1}b^N - B^{-1}Nx_N^N, \\
& \quad \Rightarrow x_B^N \approx B^{-1}b^N - \sum_{j\in\Omega} B^{-1}a_j x_j^N, \\
& \quad \Rightarrow x_B^N \approx \omega_\circ^N - \sum_{j\in\Omega} (y_j) x_j^N,
\end{aligned} \tag{4}
$$

Here, Ω indicates the current set of indices of the nonbasic variables, and $\omega_\circ^N = B^{-1}b^N, .\ y_j = B^{-1}a_j$ Furthermore, the objective value of *Eq. (1)* is given by

$$
\begin{aligned}
z^N & \approx c^N x^N \\
& \approx (c_B^N, c_N^N)\begin{pmatrix} x_B^N \\ x_N^N \end{pmatrix} \\
& \approx c_B^N x_B^N + c_N^N x_N^N. \\
\Rightarrow z^N & \approx c_B^N(\omega_\circ^N - \sum_{j\in\Omega} y_j x_j^N) + \sum_{j\in\Omega} c_j^N x_j^N. \\
\Rightarrow z^N & \approx c_B^N \omega_\circ^N - \sum_{j\in\Omega} c_B^N y_j x_j^N + \sum_{j\in\Omega} c_j^N x_j^N. \\
\Rightarrow z^N & \approx z_\circ^N - \sum_{j\in\Omega} (z_j^N - c_j^N) x_j^N,
\end{aligned} \tag{5}
$$

where $z_\circ = c_B^N \omega_\circ^N$, $z_j^N = c_B^N y_j$. In the above demonstrations, we get the following NLP:

$$
\begin{aligned}
\text{Max} \quad & z^N \approx z_\circ - \sum_{j\in\Omega} (z_j^N - c_j^N) x_j^N, \\
\text{s.t}: \quad & \sum_{j\in\Omega} \mathbf{y}_j x_j^N + \mathbf{x}_B^N \approx \omega_\circ^N, \\
& x_j^N \geq 0^N, \qquad j \in \Omega, \\
& \mathbf{x}_B^N \geq 0^N.
\end{aligned} \tag{6}
$$

4 | A New Approach to Simplex-based Methods

We first study the existing Simplex-based models and then establish our strategy. *Algorithm 1* shows the primal Simplex method in the concept of NS.

Algorithm 1. Neutrosophic primal Simplex method.

Step 1. Transform the problem into a standard form; select an initial neutrosophic BFS with the basis B; otherwise, use methods such as two-phase [68] or Big-M [69] techniques.

Step 2. By solving the neutrosophic System of linear equations, compute a neutrosophic BFS to the problem of the form $x_B^N = B^{-1}b^N = \omega_\circ^N$, $x_N^N = 0^N$ and the relevant objective function as $z^N = c_B^N \omega_\circ^N$.

Step 3. By solving the neutrosophic System of linear equations $Q^N B = c_B^N$, set $Q^N = c_B^N B^{-1}$. Then in each iteration, for each basic variable, put $z_B^N - c_B^N = 0^N$. Also for each nonbasic variable set $z_j^N - c_j^N = c_B^N y_j - c_j^N = c_B^N B^{-1} a_j - c_j^N = Q^N a_j - c_j^N$. If for all $j, z_j^N - c_j^N \geq 0^N$ the current solution is optimal, else choose $z_k^N - c_k^N = \min_{j\in\Omega}\{z_j^N - c_j^N\}$ and go to *Step 4*.

Step 4. Based on *Eq. (4)*, set $y_k = B^{-1}a_k$. If $y_k \leq 0,$ then the given problem reaches an unbounded solution, so dismiss the iteration. Otherwise, go to *Step 5*.

Step 5. Assume that x_k^N be the input variable, the leaving variable $x_{B_r}^N$ determine as

$$\frac{\omega_{r\circ}^N}{y_{rk}} = \min\left\{\frac{\omega_{i\circ}^N}{y_{ik}} : y_{ik} > 0; i = 1,...,m\right\}.$$

Step 6. Update $\omega_{i\circ}^N$ by replacing $\omega_{i\circ}^N - \frac{\omega_{r\circ}^N}{y_{rk}}$ for $i \neq r$ and $\omega_{r\circ}^N$ by $\frac{\omega_{r\circ}^N}{y_{rk}}$.

Step 7. Update $B\ \Omega$, then repeat *Steps 2-4* until reaching the optimal solution.

Step 8. Find the optimal answer and, thus, the optimal value of the objective function.

For more details and the related theorems of *Algorithm 1*, see [65], [66], [68]–[71], [73]. Although this is a good algorithm, however, it has the following flaws:

I. From *Steps 2* and *3*, this algorithm must solve some neutrosophic linear systems in all iterations.

II. From *Steps 2-7*, this algorithm involves many neutrosophic arithmetic operations.

Some researchers have extended the dual Simplex algorithm to NLP when initial neutrosophic BFS is challenging to access, but its dual BFS is available.

Based on the duality principle, *Eq. (7)* is the relative dual of the primal *Eq. (1)*

$$\begin{aligned} \text{Min} \quad & W^N \approx b^N v^N, \\ \text{s.t}: \quad & A^t v^N \geq (c^N)^t, \\ & v^N \geq 0^N, \end{aligned} \tag{7}$$

where v^N is also NN. *Algorithm 2* displays the neutrosophic dual Simplex method.

Algorithm 2. Neutrosophic dual Simplex method.

Consider,

$$\begin{aligned} \text{Min} \quad & W^N \approx c^N x^N, \\ \text{s.t}: \quad & A x^N \geq b^N, \\ & x^N \geq 0^N. \end{aligned}$$

Step 1. Transform the problem into a standard form.

Step 2. By solving the neutrosophic System of linear equations $Bx_B^N = b^N$, find a neutrosophic BFS to the problem of the form $x_B^N = B^{-1}b^N = \zeta^N$, $x_N^N = 0^N$ and the relevant objective function as $W^N = c_B^N x_B^N$.

Step 3. For all $j \in \Omega$, solve $By_j = a_j$. If for all $j \in \Omega, y_{rj} \geq 0$ then we have the infeasible solution for the primal problem and stop.

Step 4. For each basic variable, set $W_B^N - c_B^N = 0^N$, and for each nonbasic variable, solve, $\varpi B = c_B^N$, substitute $W_j^N = \varpi a_j$, and then compute $W_j^N - c_j^N$. If $\psi(\zeta^N) > 0$ the current solution is optimal, else go to *Step 4*.

Step 5. Obtain $\min(\zeta_i^N < 0^N) = \zeta_r^N (i = 1,..,m)$; so the r^th^ row is the pivot row.

Step 6. If at least one $y_{rj} < 0$, then $x_{B_r}^N$ is the leaving variable and x_k^N is an input variable with the following condition:

$$\frac{W_k^N - c_k^N}{y_{rk}} = \min_{j\in\Omega}\left\{\frac{W_j^N - c_j^N}{y_{rj}} : y_{rj} < 0\right\}.$$

Step 7. Update B, Ω, then repeat *Steps 2-6* until reaching the optimal solution.

Step 8. Find the optimal answer and, thus, the optimal value of the objective function.

For more details and the related theorems of *Algorithm 1*, see [67], [72], [74], [75]. Though this is an interesting algorithm, nevertheless it has the following flaws:

I. From *Step 2*, this algorithm must solve neutrosophic linear systems for each iteration.

II. From *Step 4*, the calculation of 'min' needs some neutrosophic operators, which can increase the complexity of the solution.

III. From *Steps 2-7*, for all iterations, this algorithm needs huge numbers of additions, subtractions, and other neutrosophic arithmetic.

We propose a practical, simple algorithm that can meaningfully decrease computational issues and handle these deficits to solve the mentioned shortcomings. We demonstrate that the optimal solution of the NLP can be obtained simply by considering the same CLP without solving any neutrosophic problems or auxiliary problems, as mentioned in [65]–[73].

Next, we present a new algorithm:

Algorithm 3. A modified version of the Neutrosophic primal/dual Simplex method

Step 1. Using the ranking function, obtain the rank of all neutrosophic parameters in NLP (*Eqs. (1)* or *(7)*) and construct the corresponding CLP.

Step 2. Solve the mentioned CLP with the standard primal/dual Simplex methods and obtain the optimal basis B_{crisp}.

Step 3. With B, and the right-hand-side vector (b^N) of the original NLP, calculate the neutrosophic optimal solution as: $x_B^N = B_{crisp}^{-1} b^N$.

Algorithm 3 uses a direct ranking function to transform the NLP into CLP. Such a transformation process must result in the loss and distortion of original NNs. However, in the following theorem, we prove that the optimal basis B crisp is equivalent to the basis B in *Eq. (1)*.

Theorem 1. The optimal solution to *Eq. (1)* based on the existing neutrosophic primal methods and *Algorithm 3* is the same.

Proof: based on *Algorithm 3*, we obtain the rank of all neutrosophic parameters in NLP (*Eq. (1)*). So, we have the following CLP:

$$\begin{aligned} \text{Max} \quad & \psi(z^N) = \psi(c^N)x^N, \\ \text{s.t:} \quad & Ax^N \le b^N, \\ & x^N \ge 0^N. \end{aligned} \tag{8}$$

Suppose that $\psi(Q^N)$ is the dual variable related to $Ax^N \le b^N$. Therefore, the dual of the *Problem (8)* is as follows:

$$\begin{aligned} \text{Min} \quad & \psi(z^N) = b^N \psi(Q^N), \\ \text{s.t:} \quad & \psi(Q^N)A \ge \psi(c^N), \\ & \psi(Q^N) \ge 0. \end{aligned} \tag{9}$$

Consequently, if B is the optimal basis of *Eq. (9)*, we have the optimal conditions of *Eq. (8)*. It should be emphasized that the *NLP (1)* and *Eq. (8)* must be solved, respectively, to produce the optimal solution using the existing neutrosophic primal approaches and our proposed algorithm. We can conclude that the results of our suggested process match those obtained using the current neutrosophic methods if we prove that both

problems have the same optimal solution. It is worth noting that the feasible space for both problems is the same. Therefore, it is sufficient to demonstrate that the optimality requirements for both issues are the same. If B is the optimal basis of *Eq. (1)*, it will be the optimal basis for the equivalent *CLP (8)*. Assume that $(x^N)^*$ is the optimal solution of *Eq. (1)* and that B is the optimal basis generated from existing neutrosophic approaches. Thus, based on *Algorithm 2*, for all $j, z_j^N - c_j^N \geq 0^N$. This circumstance is equivalent to

$$\psi(z_j^N) - \psi(c_j^N) \geq 0.$$

Moreover, according to *Step 3* of *Algorithm 1*, it is equivalent to

$$\psi(z_j^N) - \psi(c_j^N) \geq 0 \Leftrightarrow \psi(c_B^N y_j) - \psi(c_j^N) \geq 0 \Leftrightarrow$$
$$\psi(c_B^N B^{-1} a_j) - \psi(c_j^N) \geq 0 \Leftrightarrow \psi(Q^N) a_j - \psi(c_j^N) \geq 0. \Leftrightarrow \psi(Q^N) a_j \geq \psi(c_j^N).$$

Thus, we conclude that $(x^N)^*$ is the optimal solution to *Eq. (8)*, and the proof is completed.

To summarize, the following are the benefits of the proposed approach over *Algorithms 1* and *2*:

From the point of view of computational complexity, it is well known that solving a system of linear equations has a complexity of at most O (n^3), and at least n^2 operations are needed to solve a general system of n linear equations.

Based on *Algorithm 1*, to perform *Step 2* of this method, the neutrosophic System $Bx_B^N = b^N$ must be solved using m neutrosophic equations and m neutrosophic variables in all iterations. After solving this neutrosophic System, we must solve another neutrosophic system $Q^N B = c_B^N$ of linear equations in all iterations of this procedure. A minimum right-hand side test determines a system's leaving variable once these neutrosophic systems have been solved. So, based on the previously described law of complexity for linear equation systems, this calls for much computation. *Steps 3-7*, in particular, need a vast number of addition and subtraction operations on the number of NNs for all iterations, as we observe. While Algorithm 3 does not solve any neutrosophic systems or perform any neutrosophic arithmetic operations, the leaving variable can be discovered using this method. In *Algorithm 3*, all arithmetic operations are performed on real numbers, including comparing NNs.

Additionally, according to *Algorithm 2*, the neutrosophic system must be solved to proceed to *Step 2* of this algorithm. Moreover, in *Steps 3* and *4*, we must solve systems $By_j = a_j$ and $\varpi B = c_B^N$, in all iterations. Last but not least, the entry variable is chosen based on its minimum rank. The algorithm's *Steps 4-7* necessitate several neutrosophic additions and subtractions on NNs. Although *Algorithm 3* identifies the entering variable without solving any neutrosophic systems or doing any neutrosophic arithmetic operations.

These findings demonstrate that the suggested approach is more straightforward and computationally efficient than existing methods.

4 | Numerical Examples

In this part, some numerical examples will be used to verify the new method's effectiveness.

Example 1. Consider the following NLP with TNNs:

$$\text{Max } (z^N) \approx <(3,7,15);(0.4,0.5,0.7)> x_1^N + <(1,3,5);(0.8,0.2,0.1)> x_2^N,$$

subject to

$$\begin{aligned}
&6x_1^N + 4x_2^N \le\ <(3,5,6);(0.6,0.5,0.6)>,\\
&x_1^N + 2x_2^N \le\ <(5,8,10);(0.3,0.6,0.6)>,\\
&-x_1^N + x_2^N \le\ <(12,15,19);(0.6,0.4,0.5)>,\\
&\qquad x_2^N \le\ <(14,17,21);(0.8,0.2,0.6)>,\\
&x_1^N, x_2^N \ge 0^N.
\end{aligned} \tag{10}$$

If we solve this problem with the neutrosophic primal Simplex method (*Algorithm 1*), we have:

At first, we convert the problem into the standard NLP:

$$\text{Max } (z^N) \approx <(3,7,15);(0.4,0.5,0.7)> x_1^N + <(1,3,5);(0.8,0.2,0.1)> x_2^N, \tag{10}$$

subject to

$$\begin{aligned}
&6x_1^N + 4x_2^N + x_3^N =\ <(3,5,6);(0.6,0.5,0.6)>,\\
&x_1^N + 2x_2^N + x_4^N =\ <(5,8,10);(0.3,0.6,0.6)>,\\
&-x_1^N + x_2^N + x_5^N =\ <(12,15,19);(0.6,0.4,0.5)>,\\
&\qquad x_2^N + x_6^N =\ <(14,17,21);(0.8,0.2,0.6)>,\\
&x_i^N \ge 0^N (i=1,\dots,6).
\end{aligned} \tag{11}$$

Iteration one

Step 1. The initial neutrosophic BFS is $B=[a_3,a_4,a_5,a_6]$, so the NBM is

$$N=[a_1,a_2]=\begin{bmatrix} 6 & 4\\ 1 & 2\\ -1 & 1\\ 0 & 1\end{bmatrix}.$$

Step 2. Solve the neutrosophic system of linear equations $Bx_B^N = b^N$, so

$$x_B^N=\begin{bmatrix} x_3^N\\ x_4^N\\ x_5^N\\ x_6^N\end{bmatrix}=B^{-1}b^N=\begin{bmatrix} <(3,5,6);(0.6,0.5,0.6)>\\ <(5,8,10);(0.3,0.6,0.6)>\\ <(12,15,19);(0.6,0.4,0.5)>\\ <(14,17,21);(0.8,0.2,0.6)>\end{bmatrix},$$

$$x_N^N=\begin{bmatrix} x_1^N\\ x_2^N\end{bmatrix}=\begin{bmatrix} 0^N\\ 0^N\end{bmatrix},$$

$$z^N = c_B^N \omega_\circ^N = 0^N.$$

Step 3. By solving the neutrosophic system of linear equations $Q^N B = c_B^N$, we have

$$x_B^N=\begin{bmatrix} x_3^N\\ x_4^N\\ x_5^N\\ x_6^N\end{bmatrix}=B^{-1}b^N=\begin{bmatrix} <(3,5,6);(0.6,0.5,0.6)>\\ <(5,8,10);(0.3,0.6,0.6)>\\ <(12,15,19);(0.6,0.4,0.5)>\\ <(14,17,21);(0.8,0.2,0.6)>\end{bmatrix},$$

$$x_N^N=\begin{bmatrix} x_1^N\\ x_2^N\end{bmatrix}=\begin{bmatrix} 0^N\\ 0^N\end{bmatrix},$$

$$Q^N B = c_B^N \Rightarrow \left[q_1^N, q_2^N, q_3^N, q_4^N\right] B = \begin{bmatrix} 0^N \\ 0^N \\ 0^N \\ 0^N \end{bmatrix} \Rightarrow q_i^N = 0^N ; (i = 1, \ldots, 4).$$

Moreover, for each nonbasic variable ($\Omega = \{1,2\}$), compute $z_j^N - c_j^N = Q^N a_j - c_j^N$. Then

$$z_1^N - c_1^N = Q^N a_1 - c_1^N = \begin{bmatrix} 0^N & 0^N & 0^N & 0^N \end{bmatrix} \begin{bmatrix} 6 \\ 1 \\ -1 \\ 0 \end{bmatrix} - <(3,7,15);(0.4,0.5,0.7)> = <(-15,-7,-3);(0.4,0.5,0.7)>.$$

$$z_2^N - c_2^N = Q^N a_2 - c_2^N = \begin{bmatrix} 0^N & 0^N & 0^N & 0^N \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \\ 1 \end{bmatrix} - <(1,3,5);(0.8,0.2,0.1)> = <(-5,-3,-1);(0.8,0.2,0.1)>.$$

Also, for simplicity of calculation for choosing $z_k^N - c_k^N = \min\limits_{j \in \Omega} \{z_j^N - c_j^N\}$, we use the ranking function of *Definition 3*:

$$\min\{\psi(z_1^N - c_1^N), \psi(z_2^N - c_2^N)\} = \{\frac{-10}{3}, \frac{-5}{2}\} = \frac{-10}{3} \Rightarrow k = 1.$$

Step 4. Based on *Eq. (4)*, set $y_1 = B^{-1} a_1$ as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{41} \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ -1 \\ 0 \end{bmatrix} \Rightarrow y_1 = \begin{bmatrix} 6 \\ 1 \\ -1 \\ 0 \end{bmatrix}.$$

Step 5. So, based on *Step 3*, x_1^N is the input variable, and for the leaving variable, we get

$$\frac{\omega_{r\circ}^N}{y_{rk}} = \min\left\{\frac{\omega_{i\circ}^N}{y_{ik}} : y_{ik} > 0; i = 1, \ldots, m\right\} = \min\left\{\frac{\omega_{1\circ}^N}{y_{11}}, \frac{\omega_{2\circ}^N}{y_{21}}\right\} =$$
$$\min\left\{\frac{<(3,5,6);(0.6,0.5,0.6)>}{6}, \frac{<(5,8,10);(0.3,0.6,0.6)>}{1}\right\}.$$

Again, use the ranking function of *Definition 3*, $r = 1$, i.e., $x_{B_1}^N = x_3^N$ is the leaving variable.

At the end of this iteration, we must update the following parameters:

$$B = [a_1, a_4, a_5, a_6] = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\Omega = \{2,3\}.$$

Similarly, for the other iterations, we have:

Iteration two

Step 1. $B = [a_1, a_4, a_5, a_6]$, $N = [a_2, a_3]$.

Step 2. Solve $Bx_B^N = b^N$, so

$$\begin{bmatrix} 6 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^N \\ x_4^N \\ x_5^N \\ x_6^N \end{bmatrix} = \begin{bmatrix} <(3,5,6);(0.6,0.5,0.6)> \\ <(5,8,10);(0.3,0.6,0.6)> \\ <(12,15,19);(0.6,0.4,0.5)> \\ <(14,17,21);(0.8,0.2,0.6)> \end{bmatrix},$$

$$\Rightarrow \begin{bmatrix} x_1^N \\ x_4^N \\ x_5^N \\ x_6^N \end{bmatrix} = \begin{bmatrix} <(\frac{3}{6},\frac{5}{6},1);(0.6,0.5,0.6)> \\ <(4,\frac{43}{6},\frac{57}{6});(0.3,0.6,0.6)> \\ <(\frac{25}{2},\frac{95}{6},20);(0.6,0.5,0.6)> \\ <(14,17,21);(0.8,0.2,0.6)> \end{bmatrix},$$

$$x_N^N = \begin{bmatrix} x_2^N \\ x_3^N \end{bmatrix} = \begin{bmatrix} 0^N \\ 0^N \end{bmatrix},$$

$$z^N = c_B^N x_B^N =<(\frac{9}{6},\frac{35}{6},15);(0.4,0.5,0.7)>.$$

Step 3. By solving the neutrosophic System of linear equations, we have

$$\begin{bmatrix} q_1^N, q_2^N, q_3^N, q_4^N \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} <(3,7,15);(0.4,0.5,0.7)> \\ 0^N \\ 0^N \\ 0^N \end{bmatrix}$$

$$\Rightarrow \begin{cases} q_1^N =<(\frac{3}{6},\frac{7}{6},\frac{15}{6});(0.4,0.5,0.7)>, \\ q_i^N = 0^N;(i=2,...,4). \end{cases}$$

Moreover, for each nonbasic variable ($\Omega = \{2,3\}$), compute $z_j^N - c_j^N = Q^N a_j - c_j^N$. Then

$$z_2^N - c_2^N = Q^N a_2 - c_2^N =<(-3,\frac{10}{6},9);(0.4,0.5,0.7)>,$$

$$z_3^N - c_3^N = Q^N a_3 - c_3^N =<(\frac{3}{6},\frac{7}{6},\frac{15}{6});(0.4,0.5,0.7)>.$$

$k = 2.$

Step 4. Based on *Eq. (4)*, set $y_2 = B^{-1}a_2$ as

$$\begin{bmatrix} 6 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{12} \\ y_{22} \\ y_{32} \\ y_{42} \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 1 \\ 1 \end{bmatrix} \Rightarrow y_1 = \begin{bmatrix} \frac{2}{3} \\ \frac{4}{3} \\ \frac{5}{3} \\ 1 \end{bmatrix}.$$

Step 5. So, based on *Step 3*, x_2^N is the input variable, and for the leaving variable, we get

$$\frac{\omega_{r\circ}^N}{y_{rk}} = \min\left\{\frac{\omega_{i\circ}^N}{y_{ik}} : y_{ik} > 0; i = 1,\ldots,m\right\} = \min\left\{\frac{\omega_{1\circ}^N}{y_{12}}, \frac{\omega_{2\circ}^N}{y_{22}}, \frac{\omega_{3\circ}^N}{y_{32}}, \frac{\omega_{4\circ}^N}{y_{42}}\right\}.$$

Again, use the ranking function of *Definition 3*, $r = 1$, i.e., $x_{B_1}^N = x_1^N$ is the leaving variable.

At the end of this iteration, we must update the following parameters:

$$B = [a_2, a_4, a_5, a_6] = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

$$\Omega = \{1,3\}.$$

Moreover, for the next iteration:

Iteration three

Step 1. $B = [a_2, a_4, a_5, a_6]$, $N = [a_1, a_3]$.

Step 2. Solve $Bx_B^N = b^N$, so

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_2^N \\ x_4^N \\ x_5^N \\ x_6^N \end{bmatrix} = \begin{bmatrix} <(3,5,6);(0.6,0.5,0.6)> \\ <(5,8,10);(0.3,0.6,0.6)> \\ <(12,15,19);(0.6,0.4,0.5)> \\ <(14,17,21);(0.8,0.2,0.6)> \end{bmatrix},$$

$$\Rightarrow \begin{bmatrix} x_2^N \\ x_4^N \\ x_5^N \\ x_6^N \end{bmatrix} = \begin{bmatrix} <(\frac{3}{4},\frac{5}{4},\frac{6}{4});(0.6,0.5,0.6)> \\ <(11,18,22);(0.3,0.6,0.6)> \\ <(15,20,25);(0.6,0.5,0.6)> \\ <(17,22,27);(0.6,0.5,0.6)> \end{bmatrix},$$

$$x_N^N = \begin{bmatrix} x_1^N \\ x_3^N \end{bmatrix} = \begin{bmatrix} 0^N \\ 0^N \end{bmatrix},$$

$$z^N = c_B^N x_B^N = <(\frac{3}{4},\frac{15}{4},\frac{30}{4});(0.6,0.5,0.6)>.$$

Step 3. By solving the neutrosophic System of linear equations, we have

$$\left[q_1^N, q_2^N, q_3^N, q_4^N\right]\begin{bmatrix} 4 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} <(1,3,5);(0.8,0.2,0.1)> \\ 0^N \\ 0^N \\ 0^N \end{bmatrix}$$

$$\Rightarrow \begin{cases} q_1^N = <(\frac{1}{4},\frac{3}{4},\frac{5}{4});(0.8,0.2,0.1)>, \\ q_i^N = 0^N; (i = 2,\ldots,4). \end{cases}$$

Moreover, for each nonbasic variable ($\Omega = \{1,3\}$), compute $z_j^N - c_j^N = Q^N a_j - c_j^N$. Then

$$z_1^N - c_1^N = Q^N a_1 - c_1^N =< (\frac{3}{2}, \frac{5}{2}, \frac{9}{2}); (0.4, 0.5, 0.7) >,$$

$$z_3^N - c_3^N = Q^N a_3 - c_3^N =< (\frac{1}{4}, \frac{3}{4}, \frac{5}{4}); (0.8, 0.2, 0.1) > .$$

Since for all $j, z_j^N - c_j^N \geq 0^N$, based on *Step 3* on *Algorithm 1*, the current solution is optimal. So

$$(x_B^N)^* = \begin{bmatrix} (x_2^N)^* \\ (x_4^N)^* \\ (x_5^N)^* \\ (x_6^N)^* \end{bmatrix} = \begin{bmatrix} < (\frac{3}{4}, \frac{5}{4}, \frac{6}{4}); (0.6, 0.5, 0.6) > \\ < (11, 18, 22); (0.3, 0.6, 0.6) > \\ < (15, 20, 25); (0.6, 0.5, 0.6) > \\ < (17, 22, 27); (0.6, 0.5, 0.6) > \end{bmatrix},$$

$$(z^N)^* =< (\frac{3}{4}, \frac{15}{4}, \frac{30}{4}); (0.6, 0.5, 0.6) > .$$

With the new method, we solve the *NLP (10)* much easier than the above method.

Step 1. Using the ranking function of *Definition 3*, we get the rank of all neutrosophic parameters and construct the following CLP:

$$\text{Max } (z) = \frac{10}{3} x_1 + \frac{5}{2} x_2,$$

$$6x_1 + 4x_2 \leq \frac{7}{3},$$

$$x_1 + 2x_2 \leq \frac{253}{90},,$$

$$-x_1 + x_2 \leq \frac{391}{45},$$

$$x_2 \leq \frac{104}{9},$$

$$x_1, x_2 \geq 0.$$

By Geometric methods such as *Fig. 1*, it is easy to see that the optimal solution of LP of *Step 1* is $x_1 = 0$, and $x_2 = \frac{7}{12}$.

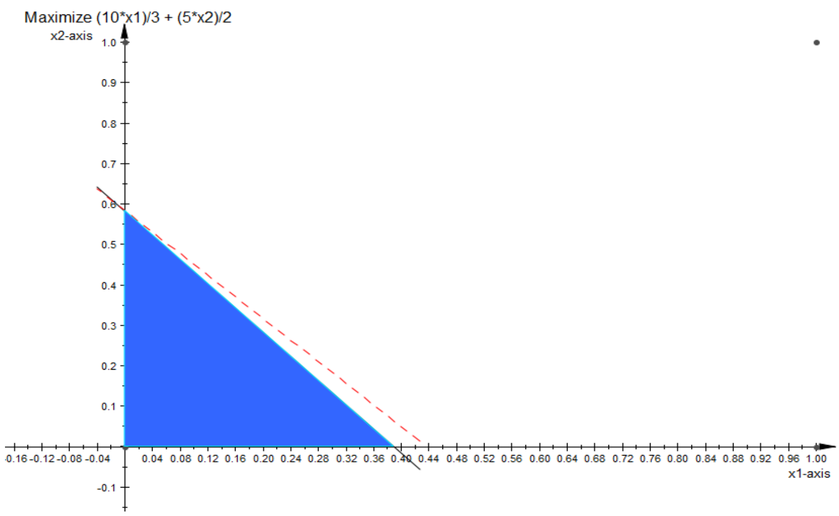

Fig. 1. The graphical show of Step 1.

Step 2. By solving the standard primal simplex method, we obtain $B_{crisp} = [a_2, a_4, a_5, a_6]$ as the optimal basis. So

$$B_{crisp}^{-1} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ \frac{-1}{2} & 1 & 0 & 0 \\ \frac{-1}{4} & 0 & 1 & 0 \\ \frac{-1}{4} & 0 & 0 & 1 \end{bmatrix}.$$

Step 3. Finally, we compute the neutrosophic optimal solution as

$$x_B^N = \begin{bmatrix} x_2^N \\ x_4^N \\ x_5^N \\ x_6^N \end{bmatrix} = B_{crisp}^{-1} b^N = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ \frac{-1}{2} & 1 & 0 & 0 \\ \frac{-1}{4} & 0 & 1 & 0 \\ \frac{-1}{4} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} <(3,5,6);(0.6,0.5,0.6)> \\ <(5,8,10);(0.3,0.6,0.6)> \\ <(12,15,19);(0.6,0.4,0.5)> \\ <(14,17,21);(0.8,0.2,0.6)> \end{bmatrix} =$$

$$\begin{bmatrix} (\frac{3}{4}, \frac{5}{4}, \frac{6}{4});(0.6,0.5,0.6) \\ (2, \frac{11}{2}, \frac{17}{2});(0.3,0.6,0.6) \\ (\frac{21}{2}, \frac{55}{4}, \frac{73}{4});(0.6,0.5,0.6) \\ (\frac{25}{2}, \frac{63}{4}, \frac{81}{4});(0.6,0.5,0.6) \end{bmatrix},$$

And $(Z^N)^* =< (\frac{3}{4}, \frac{15}{4}, \frac{30}{4});(0.6,0.5,0.6) >$.

Example 2 ([69]). Consider the following NLP with SVTNNs:

$$\text{Min } (W^N) \approx< 10c_1^N x_1^N + 10c_2^N x_2^N,$$

subject to

$$4x_1^N + 7x_2^N \geq 10b_1^N,$$
$$3x_1^N + 5x_2^N \geq 10b_2^N,$$
$$x_1^N, x_2^N \geq 0^N,$$

where

$$b_1^N =< [4,1,1],[4,3,4],[4,2,9] >,$$
$$b_2^N =< [11,8,4],[11,6,3],[11,2,5] >,$$
$$c_1^N =< [5,4,3],[5,3,4],[5,1,2] >,$$
$$c_1^N =< [6,4,6],[6,2,5],[6,3,4] >.$$

This problem is solved in [74] by the dual Simplex method (*Algorithm 2*), where it can be seen that the problem-solving process is complex and full of neutrosophic arithmetic operations.

Next, we solve this problem by our method:

Step 1. Using the ranking function of *Definition 5* ($\alpha = 0.6$), we get the rank of all neutrosophic parameters and construct the following CLP:

$$\text{Min (W)} = \frac{335}{9}x_1 + \frac{418}{9}x_2,$$

s.t.

$$\begin{aligned}
&4x_1 + 7x_2 \geq \frac{296}{9},\\
&3x_1 + 5x_2 \geq \frac{238}{3},\\
&x_1, x_2 \geq 0.
\end{aligned}$$

Step 2. The standard form of this problem is as follows:

$$\text{Min (W)} = \frac{335}{9}x_1 + \frac{418}{9}x_2,$$

s.t.

$$\begin{aligned}
&-4x_1 - 7x_2 + x_3 = \frac{-296}{9},\\
&-3x_1 - 5x_2 + x_4 = \frac{-238}{3},\\
&x_i \geq 0 (i = 1,\dots,4).
\end{aligned}$$

By solving the standard dual simplex method, we obtain $B_{crisp} = [a_2, a_3] = \begin{bmatrix} 7 & -1 \\ 5 & 0 \end{bmatrix}$ as the optimal basis. So

$$B_{crisp}^{-1} = \begin{bmatrix} 0 & \frac{1}{5} \\ -1 & \frac{7}{5} \end{bmatrix}.$$

Step 3. Hence, the optimal solution has arrived with the value of variables as

$$x_B^N = \begin{bmatrix} x_2^N \\ x_3^N \end{bmatrix} = B_{crisp}^{-1} b^N = \begin{bmatrix} 0 & \frac{1}{5} \\ -1 & \frac{7}{5} \end{bmatrix} \begin{bmatrix} 10b_1^N \\ 10b_2^N \end{bmatrix} = \begin{bmatrix} 2b_2^N \\ 14b_2^N - 10b_1^N \end{bmatrix},$$

and

$$(W^N)^* = (c_B^N)(x_B^N) = (10c_2^N)(2b_2^N) = 20c_2^N b_2^N.$$

This is the same as the solution obtained from the existing method [74], but achieving this answer with a new perspective is much easier and with minimal neutrosophic arithmetic operations.

6 | Conclusion

This paper introduced LP problems with NNs and studied the existing SBMs (primal and dual) in this environment. We expressed the disadvantages and challenges of these methods. We then proposed a simple and efficient approach to solving these problems that can overcome the shortcomings of existing processes and significantly reduce the complexity of the problem. The numerical results confirmed this claim and showed that the new approach could promise its practical application in real-world problems and effectively solve other optimization problems in the neutrosophic environment. It is worth noting that the uncertainty and indeterminacy in this paper are limited to single-valued NNs. However, different types of NNs, such as bipolar NSs and interval-valued NSs, hesitant NSs, etc., can also represent nucleus-specific variables in the mentioned problem. We intend to extend the proposed approach to these types of tools in the future.

Conflicts of Interest

The authors declare they have no conflicts of interest to report regarding the present study.

Acknowledgments

The author would like to thank the referees for their valuable suggestions and helpful comments in improving this paper.

Author Contribution

All authors contributed equally to this work. All authors read and approved the final manuscript.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

References

[1] Zimmermann, H. J. (1975). Description and optimization of fuzzy systems. *International journal of general system*, *2*(1), 209–215. https://doi.org/10.1080/03081077508960870

[2] Tanaka, H., & Asai, K. (1984). Fuzzy linear programming problems with fuzzy numbers. *Fuzzy sets and systems*, *13*(1), 1–10. DOI:10.1016/0165-0114(84)90022-8

[3] Verdegay, J. L. (1984). Applications of fuzzy optimization in operational research. *Control and cybernetics*, *13*(3), 229–239.

[4] Najafi, H. S., & Edalatpanah, S. A. (2014). Homotopy perturbation method for linear programming problems. *Applied mathematical modelling*, *38*(5–6), 1607–1611. https://doi.org/10.1016/j.apm.2013.09.011

[5] Delgado, M., Verdegay, J. L., & Vila, M. A. (1989). A general model for fuzzy linear programming. *Fuzzy sets and systems*, *29*(1), 21–29. https://doi.org/10.1016/0165-0114(89)90133-4

[6] Edalatpanah, S. A. (2023). Multidimensional solution of fuzzy linear programming. *Peerj computer science*, *9*, e1646. https://doi.org/10.7717/peerj-cs.1646

[7] Das, S. K., Edalatpanah, S. A., & Mandal, T. (2021). Development of unrestricted fuzzy linear fractional programming problems applied in real case. *Fuzzy information and engineering*, *13*(2), 184–195. https://doi.org/10.1080/16168658.2021.1915553

[8] Behera, D., Peters, K., Edalatpanah, S. A., & Qiu, D. (2021). New methods for solving imprecisely defined linear programming problem under trapezoidal fuzzy uncertainty. *Journal of information and optimization sciences*, *42*(3), 603–629. https://doi.org/10.1080/02522667.2020.1758369

[9] Pérez-Cañedo, B., Concepción-Morales, E. R., & Edalatpanah, S. A. (2019). A revised version of a lexicographical-based method for solving fully fuzzy linear programming problems with inequality

constraints. *Fuzzy information and engineering*, *11*(4), 474–493. https://doi.org/10.1080/16168658.2020.1761511

[10] Edalatpanah, S. A. (2023). A paradigm shift in linear programming: an algorithm without artificial variables. *Systemic analytics*, *1*(1), 1–10. https://doi.org/10.31181/sa1120232

[11] Hosseinzadeh, A., & Edalatpanah, S. (2016). A new approach for solving fully fuzzy linear programming by using the lexicography method. *Advances in fuzzy systems*, *2016*(1), 1538496. https://doi.org/10.1155/2016/1538496

[12] Das, S. K., Edalatpanah, S. A., & Mandal, T. (2018). A proposed model for solving fuzzy linear fractional programming problem: numerical point of view. *Journal of computational science*, *25*, 367–375. https://doi.org/10.1016/j.jocs.2017.12.004

[13] Najafi, H. S., Edalatpanah, S. A., & Dutta, H. (2016). A nonlinear model for fully fuzzy linear programming with fully unrestricted variables and parameters. *Alexandria engineering journal*, *55*(3), 2589–2595. https://doi.org/10.1016/j.aej.2016.04.039

[14] Das, S. K., Mandal, T., & Edalatpanah, S. A. (2017). A new approach for solving fully fuzzy linear fractional programming problems using the multi-objective linear programming. *RAIRO-operations research*, *51*(1), 285–297. https://doi.org/10.1051/ro/2016022

[15] Najafi, H. S., & Edalatpanah, S. A. (2013). A note on "A new method for solving fully fuzzy linear programming problems." *Applied mathematical modelling*, *37*(14–15), 7865–7867. https://doi.org/10.1016/j.apm.2013.02.039

[16] Das, S. K., Mandal, T., & Edalatpanah, S. A. (2017). A mathematical model for solving fully fuzzy linear programming problem with trapezoidal fuzzy numbers. *Applied intelligence*, *46*, 509–519. https://doi.org/10.1007/s10489-016-0779-x%0A%0A

[17] Wan, S. P., Wang, F., Lin, L. L., & Dong, J. Y. (2015). An intuitionistic fuzzy linear programming method for logistics outsourcing provider selection. *Knowledge-based systems*, *82*, 80–94. https://doi.org/10.1016/j.knosys.2015.02.027

[18] Pérez-Cañedo, B., & Concepción-Morales, E. R. (2019). On LR-type fully intuitionistic fuzzy linear programming with inequality constraints: Solutions with unique optimal values. *Expert systems with applications*, *128*, 246–255. https://doi.org/10.1016/j.eswa.2019.03.035

[19] Aliev, R. A., Alizadeh, A. V, Huseynov, O. H., & Jabbarova, K. I. (2015). Z-number-based linear programming. *International journal of intelligent systems*, *30*(5), 563–589. https://doi.org/10.1002/int.21709

[20] Ahmad, F., & Adhami, A. Y. (2021). Spherical fuzzy linear programming problem. In *Decision making with spherical fuzzy sets: theory and applications* (pp. 455–472). Springer.

[21] Chen, Z., Chen, Q., Chen, W., & Wang, Y. (2004). Grey linear programming. *Kybernetes*, *33*(2), 238–246. https://doi.org/10.1108/03684920410514166

[22] Darvishi, D., Liu, S., & Yi-Lin Forrest, J. (2021). Grey linear programming: a survey on solving approaches and applications. *Grey systems: theory and application*, *11*(1), 110–135. https://doi.org/10.1108/GS-04-2020-0043

[23] Hamzehee, A., Yaghoobi, M. A., & Mashinchi, M. (2014). Linear programming with rough interval coefficients. *Journal of intelligent & fuzzy systems*, *26*(3), 1179–1189. 10.3233/IFS-130804

[24] Lu, H., Huang, G., & He, L. (2011). An inexact rough-interval fuzzy linear programming method for generating conjunctive water-allocation strategies to agricultural irrigation systems. *Applied mathematical modelling*, *35*(9), 4330–4340. https://doi.org/10.1016/j.apm.2011.03.008

[25] Vasant, P., Nagarajan, R., & Yaacob, S. (2005). Fuzzy linear programming with vague objective coefficients in an uncertain environment. *Journal of the operational research society*, *56*(5), 597–603. https://doi.org/10.1057/palgrave.jors.2601855

[26] Nedoma, J. (1993). Vague matrices in linear programming. *Annals of operations research*, *46*, 483–496. https://doi.org/10.1007/BF02023110%0A%0A

[27] Ranjbar, M., & Effati, S. (2019). Symmetric and right-hand-side hesitant fuzzy linear programming. *IEEE transactions on fuzzy systems*, *28*(2), 215–227. https://doi.org/10.1109/TFUZZ.2019.2902109

[28] Farnam, M., & Darehmiraki, M. (2021). Solution procedure for multi-objective fractional programming problem under hesitant fuzzy decision environment. *Journal of fuzzy extension and applications*, *2*(4), 364–376. https://doi.org/10.22105/jfea.2021.288198.1152

[29] Akram, M., Ullah, I., Allahviranloo, T., & Edalatpanah, S. A. (2021). LR-type fully Pythagorean fuzzy linear programming problems with equality constraints. *Journal of intelligent & fuzzy systems*, *41*(1), 1975–1992. DOI:10.3233/JIFS-210655

[30] Akram, M., Ullah, I., & Alharbi, M. G. (2021). Methods for solving LR-Type Pythagorean fuzzy linear programming problems with mixed constraints. *Mathematical problems in engineering*, *2021*(1), 4306058. https://doi.org/10.1155/2021/4306058

[31] Das, S. K., & Edalatpanah, S. A. (2021). Application of linear programming in diet problem under Pythagorean fuzzy environment. In *Pythagorean fuzzy sets: theory and applications* (pp. 315–327). Springer. https://doi.org/10.1007/978-981-16-1989-2_13%0A%0A

[32] Akram, M., Ullah, I., Allahviranloo, T., & Edalatpanah, S. A. (2021). Fully Pythagorean fuzzy linear programming problems with equality constraints. *Computational and applied mathematics*, *40*(4), 1–30.

[33] Smarandache, F. (1999). A unifying field in logics: neutrosophic logic. In *Philosophy* (pp. 1–141). American Research Press.

[34] Ye, J. (2014). A multicriteria decision-making method using aggregation operators for simplified neutrosophic sets. *Journal of intelligent & fuzzy systems*, *26*(5), 2459–2466. DOI:10.3233/IFS-130916

[35] Ye, J. (2017). Subtraction and division operations of simplified neutrosophic sets. *Information*, *8*(2), 51. https://doi.org/10.3390/info8020051

[36] Edalatpanah, S. A., & Smarandache, F. (2019). *Data envelopment analysis for simplified neutrosophic sets*. Infinite Study.

[37] Wang, H., Smarandache, F., Sunderraman, R., & Zhang, Y. Q. (2005). *Interval neutrosophic sets and logic: theory and applications in computing: Theory and applications in computing* (Vol. 5). Infinite Study.

[38] Kakati, P. (2021). Interval Neutrosophic Einstein prioritized normalized weighted geometric Bonferroni mean operator and its application to multicriteria decision making. *Neural processing letters*, *53*(5), 3395–3425. https://doi.org/10.1007/s11063-021-10553-3%0A%0A

[39] Garai, T., & Garg, H. (2022). Possibilistic multiattribute decision making for water resource management problem under single-valued bipolar neutrosophic environment. *International journal of intelligent systems*, *37*(8), 5031–5058.

[40] Stanujkic, D., Karabasevic, D., Popovic, G., Smarandache, F., Zavadskas, E. K., Meidute-Kavaliauskiene, I., & Ulutaş, A. L. P. T. E. K. İ. N. (2021). Developing a novel approach for determining the reliability of bipolar neutrosophic sets and its application in multi-criteria decision-making. *Journal of multiple-valued logic & soft computing*, *37*, 151. https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=15423980&AN=152390014&h=VOrO%2BdsWxsbWEtUhESmeA7bbRpCmk7WXe%2B%2Fk3jMVpA1eJ%2BgCVpDxjqQd7wSXuMK%2BIo8GKETxOx05FmYlVBhw7A%3D%3D&crl=c

[41] Edalatpanah, S. A. (2020). Neutrosophic structured element. *Expert systems*, *37*(5), e12542. https://doi.org/10.1111/exsy.12542

[42] Bhat, S. A. (2023). An enhanced AHP group decision-making model employing neutrosophic trapezoidal numbers. *Journal operation strateght analysis*, *1*(2), 81–89. DOI:10.56578/josa010205

[43] Saberhoseini, S. F., Edalatpanah, S. A., & Sorourkhah, A. (2022). Choosing the best private-sector partner according to the risk factors in neutrosophic environment. *Big data and computing visions*, *2*(2), 61–68. DOI:10.22105/bdcv.2022.334005.1075

[44] Mohanta, K. K., & Toragay, O. (2023). Enhanced performance evaluation through neutrosophic data envelopment analysis leveraging pentagonal neutrosophic numbers. *Journal of operational and strategic analytics*, *1*(2), 70–80. https://doi.org/10.56578/josa010204

[45] Edalatpanah, S. A., Hassani, F. S., Smarandache, F., Sorourkhah, A., Pamucar, D., & Cui, B. (2024). A hybrid time series forecasting method based on neutrosophic logic with applications in financial issues. *Engineering applications of artificial intelligence*, *129*, 107531. DOI:10.1016/j.engappai.2023.107531

[46] Ihsan, M., Saeed, M., & Rahman, A. U. (2023). Optimizing hard disk selection via a fuzzy parameterized single-valued neutrosophic soft set approach. *Journal operation strategic analysis*, *1*(2), 62–69. https://doi.org/10.56578/josa010203

[47] Nafei, A., Azizi, S. P., Edalatpanah, S. A., & Huang, C.-Y. (2024). Smart TOPSIS: A neural Network-Driven TOPSIS with neutrosophic triplets for green supplier selection in sustainable manufacturing. *Expert systems with applications*, *255*, 124744. https://doi.org/10.1016/j.eswa.2024.124744

[48] Mohamed, M., Salam, A., & Ye, J. (2024). Selection of sustainable material for the construction of drone aerodynamic wing using Neutrosophic RAWEC. *Systems assessment and engineering management*, *1*, 54–72. https://doi.org/10.61356/j.saem.2024.1295

[49] Mostafa, N. N., Kumar, A. K., & Ali, Y. (2024). A comparative study on X-Ray image enhancement based on neutrosophic set. *Sustainable machine intelligence journal*, *7*, 1–2. https://doi.org/10.61356/SMIJ.2024.77102

[50] Abdullah, W. (2024). A study of neutrosophic sets and deep learning models for breast cancer classification. *Multicriteria algorithms with applications*, *3*, 50–59.

[51] Das, P., & Roy, T. K. (2015). Multi-objective non-linear programming problem based on neutrosophic optimization technique and its application in riser design problem. *Neutrosophic sets and systems*, *9*(1), 15. https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=1102&context=nss_journal

[52] Hussian, A.-N., Mohamed, M., Abdel-Baset, M., & Smarandache, F. (2017). *Neutrosophic linear programming problems*. Infinite Study.

[53] Ye, J. (2018). Neutrosophic number linear programming method and its application under neutrosophic number environments. *Soft computing*, *22*, 4639–4646. https://doi.org/10.1007/s00500-017-2646-z%0A%0A

[54] Abdel-Basset, M., Gunasekaran, M., Mohamed, M., & Smarandache, F. (2019). A novel method for solving the fully neutrosophic linear programming problems. *Neural computing and applications*, *31*, 1595–1605. https://doi.org/10.1007/s00521-018-3404-6%0A%0A

[55] Singh, A., Kumar, A., & Appadoo, S. S. (2019). A novel method for solving the fully neutrosophic linear programming problems: Suggested modifications. *Journal of intelligent & fuzzy systems*, *37*(1), 885–895. DOI:10.3233/JIFS-181541

[56] Kumar, R., Edalatpanah, S. A., Gayen, S., & Broumi, S. (2021). Answer note "a novel method for solving the fully neutrosophic linear programming problems: suggested modifications." *Neutrosophic sets and systems*, *39*, 147. https://books.google.com/books?hl=en&lr=&id=CFxCEAAAQBAJ&oi=fnd&pg=PA147&dq=1.%09Kumar,+R.,+Edalatpanah,+S.+A.,+Gayen,+S.,+%26+Broum,+S.+(2021).+Answer+Note+%22A+novel+method+for+solving+the+fully+neutrosophic+linear+programming+problems:+Suggested+modifications%22.+Neutrosophic+sets+and+systems,+39(1),+148-152.&ots=NzdYGBs_Oa&sig=cfXAjdOi-422MJOfyE6tWH5D1yo

[57] Edalatpanah, S. A. (2019). A nonlinear approach for neutrosophic linear programming. *Journal of applied research on industrial engineering*, *6*(4), 367–373. https://doi.org/10.22105/jarie.2020.217904.1137

[58] Edalatpanah, S. A., & others. (2020). A direct model for triangular neutrosophic linear programming. *International journal of neutrosophic science*, *1*(1), 19–28. DOI:10.5281/zenodo.3679499 19

[59] Khatter, K. (2020). Neutrosophic linear programming using possibilistic mean. *Soft computing*, *24*(22), 16847–16867. https://doi.org/10.1007/s00500-020-04980-y%0A%0A

[60] Das, S. K., & Dash, J. K. (2020). *Modified solution for neutrosophic linear programming problems with mixed constraints*. Infinite Study.

[61] Khalifa, H. A. E. W., & Kumar, P. (2020). Solving fully neutrosophic linear programming problem with application to stock portfolio selection. *Croatian operational research review*, *11*(2), 165–176. DOI:10.17535/CRORR.2020.0014

[62] Das, S. K., Edalatpanah, S. A., & Dash, J. K. (2020). An intelligent dual simplex method to solve triangular neutrosophic linear fractional programming problem. In *Neutrosophic sets and systems* (Vol. 36, pp. 50–69). Infinite Study.

[63] Abdelfattah, W. (2021). A parametric approach to solve neutrosophic linear programming models. *Journal of information and optimization sciences*, *42*(3), 631–654. https://doi.org/10.1080/02522667.2020.1764695

[64] Ahmed, J. (2021). LR-type fully single-valued neutrosophic linear programming problems. *Neutrosophic sets and systems*, *46*(1), 416–444.

[65] Bera, T., & Mahapatra, N. K. (2019). Generalised single valued neutrosophic number and its application to neutrosophic linear programming. In *Neutrosophic sets and systems* (pp. 85–103). Infinite Study.

[66] Bera, T., Mahapatra, N. K., & others. (2020). An approach to solve the linear programming problem using single valued trapezoidal neutrosophic number. *International journal of neutrosophic science*, *3*(2), 54–66. DOI:10.5281/zenodo.3740647

[67] Bera, T., & Mahapatra, N. K. (2019). Optimisation by dual simplex approach in neutrosophic environment. *International journal of fuzzy computation and modelling*, *2*(4), 334–352. https://doi.org/10.1504/IJFCM.2019.100347

[68] Badr, E. M., Salam, M. A., & Smarandache, F. (2020). Solving neutrosophic linear programming problems with two-phase approach. In *Neutrosophic sets in decision analysis and operations research* (pp. 391–412). IGI Global.

[69] Bera, T., & Mahapatra, N. K. (2020). Neutrosophic linear programming problem and its application to real life. *Africa mathematical*, *31*(3), 709–726. https://doi.org/10.1007/s13370-019-00754-4%0A%0A

[70] Nafei, A., Wenjun, Y., & Nasseri, H. (2020). A new method for solving interval neutrosophic linear programming problems. *Gazi university journal of science*, *33*(4), 796–808. https://doi.org/10.35378/gujs.689125

[71] El Seedy, E., Elrayes, A., Badr, E., & Rabie, A. (2021). A novel artificial variable-free simplex algorithm for solving neutrosophic linear programming problems. *Italian journal of pure and applied mathematics*, *46*, 941–958.

[72] Rabie, A., el Seidy, E., Elrayes, A., & Badr, E. (2021). Dual artificial variable-free simplex algorithm for solving neutrosophic linear programming problems. *Neutrosophic sets and systems*, *46*, 36–49.

[73] Kar, R., Shaw, A. K., & Das, B. (2021). Neutrosophic fuzzy number and its application in solving linear programming problem by simplex algorithm: An alternative approach. *Annals of optimization theory and practice*, *4*(1), 29–39. https://doi.org/10.22121/aotp.2021.278103.1064

[74] Bera, T., & Mahapatra, N. K. (2021). On solving linear programming problem by duality approach in neutrosophic environment. *International journal of mathematics in operational research*, *18*(3), 310–335. https://doi.org/10.1504/IJMOR.2021.113576

[75] Khalid, H. E., & Essa, A. K. (2021). The duality approach of the neutrosophic linear programming. *Neutrosophic sets and systems*, *46*, 9–23.